

SuperSSD: Rethinking JBOF to Break the Cost-Performance Tradeoff in All-Flash Storage

Wenhua Wang, Hao Xiao, Heqing Tang, Bin Zhang,

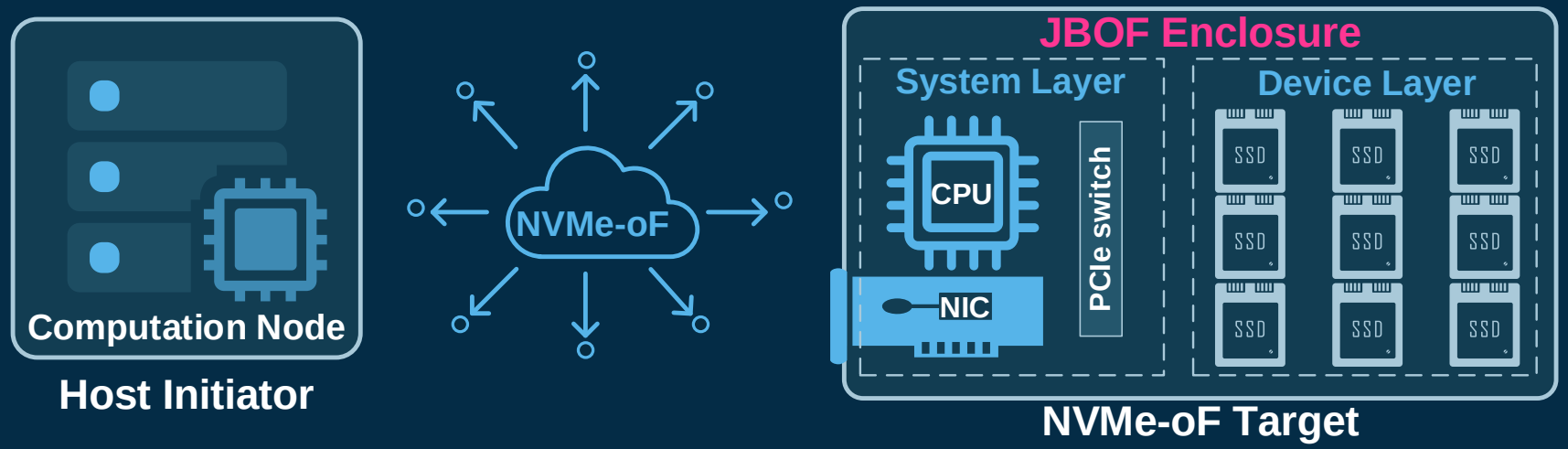
Xingyu Wang, Luoqing Zhou, Kaiyi Wen

Flyslice Technologies · Hefei University of Technology · Altera



JBOF for Data-Intensive Computing

AI and data-intensive workloads need efficient access to data.



Compute clients

Access pooled flash remotely

JBOF: Just a Bunch of Flash

Pooled SSDs, separate from compute
NVMe-over-Fabrics access
Flexible capacity and resource sharing

Performance and Cost Pressures

Performance pressure

Protocol processing adds latency.
Under concurrency, the system layer can bottleneck the SSD array.

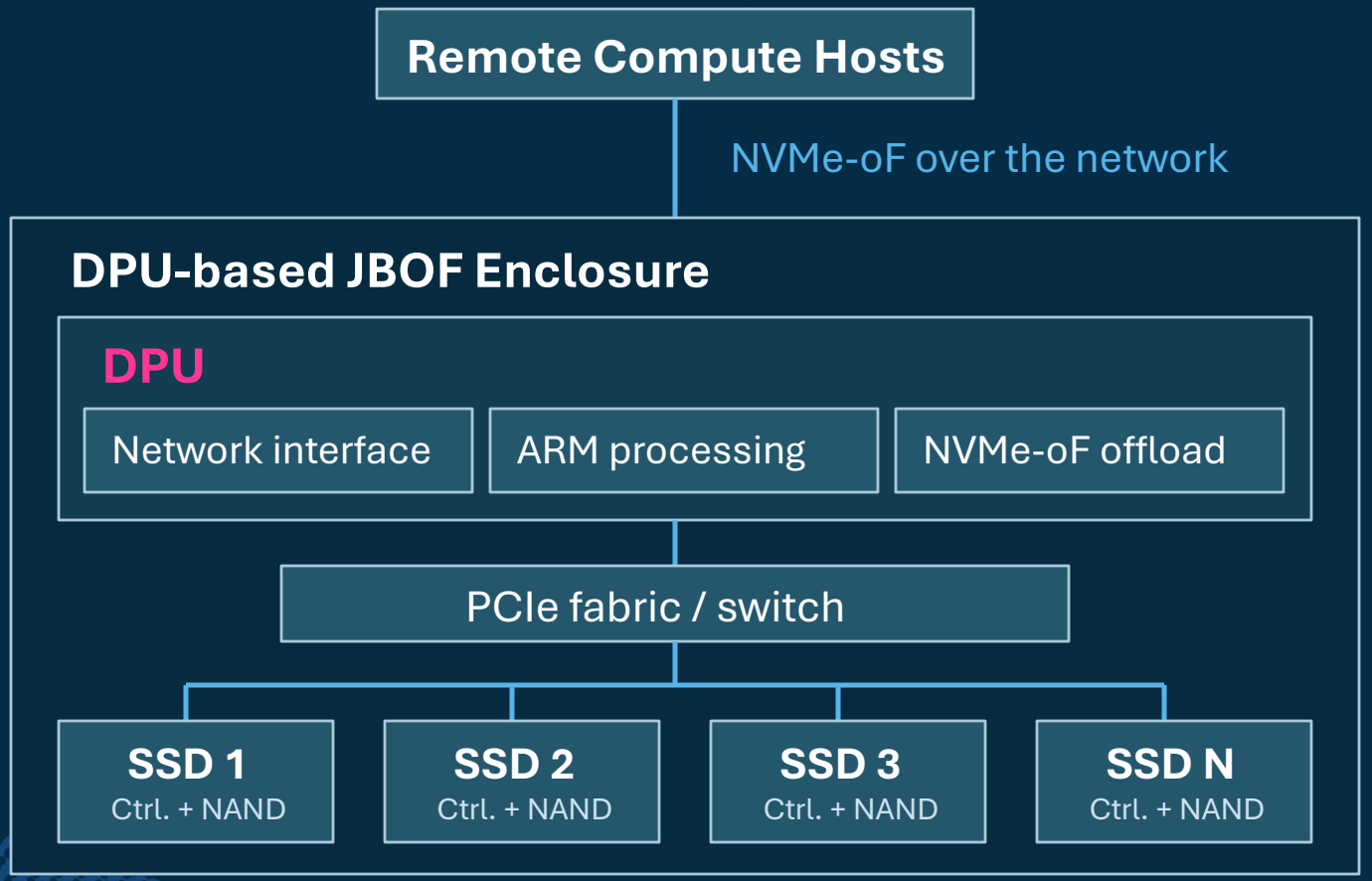
Cost pressure

Petabyte-scale flash needs many SSDs.
Cheaper drives trade cost for weaker random I/O and error protection.

Objective

Lower enclosure cost while sustaining high performance and reliable data access.

DPU Offload Addresses Only Part of the Problem



Conceptual architecture. DPU integration and PCIe topology vary by implementation.

DPU benefit

Protocol offload reduces system processing overhead.

Device layer unchanged

Each SSD still provides its own FTL, GC, and ECC.

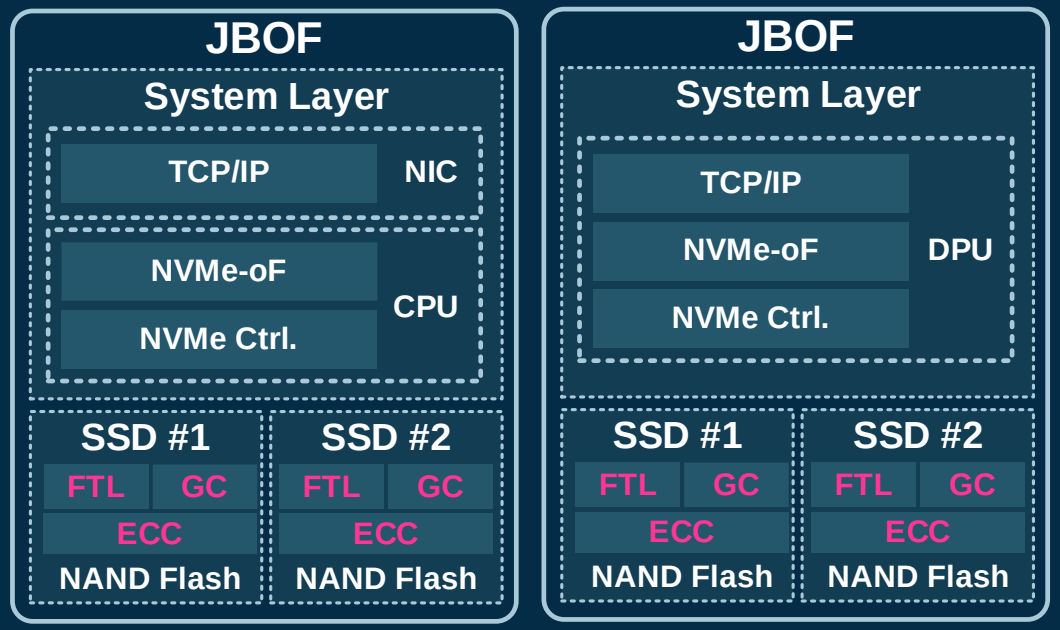
~80% of JBOF cost

comes from SSDs. DPU offload leaves this dominant cost largely untouched.

SC26

The Conventional System–Device Boundary

A block interface leaves NAND management inside each SSD.



The same controller capabilities repeat across the SSD array.

ECC: correction at high throughput

Strong error correction requires complex ECC logic, computation, and power.

FTL / GC: fast flash management

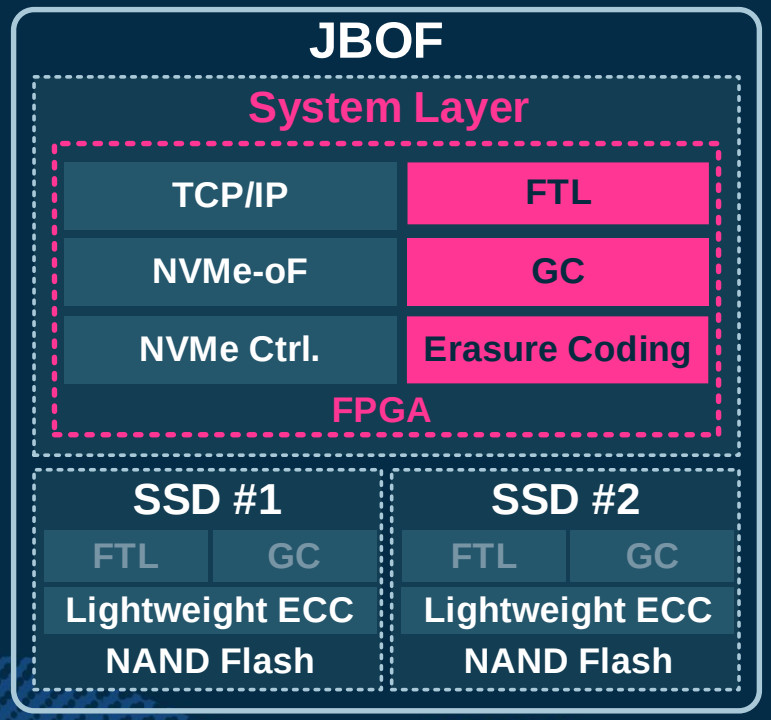
Mapping and garbage collection require controller processing and mapping memory.

Combined: costly per-drive controllers

Each SSD needs substantial resources to meet performance and reliability targets.

Our Idea: Coordinate the Enclosure and Its SSDs

Treat the SSD array as one coordinated system.



Shared FTL and aggregation

Organize random writes and simplify device mapping.

Cross-drive erasure coding

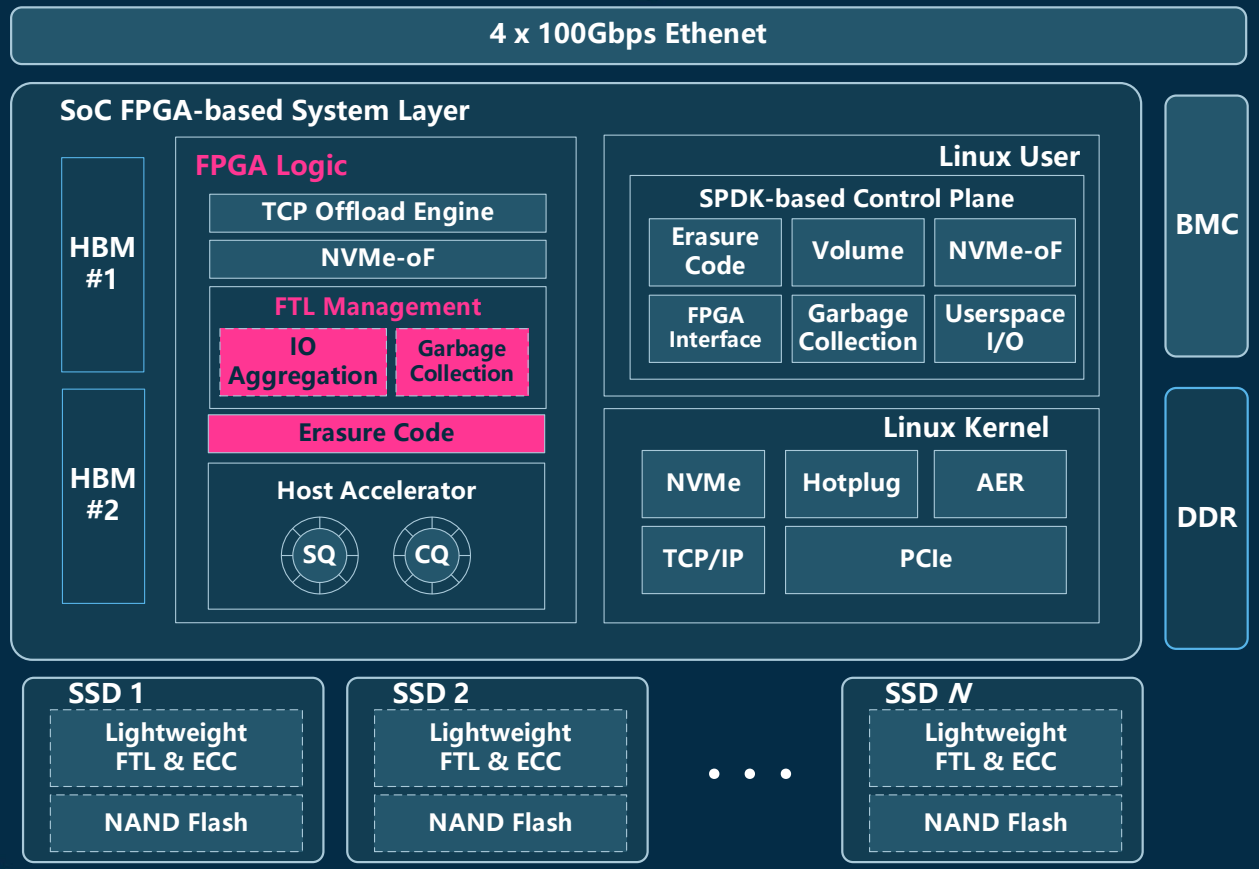
Add system protection and exploit aggregate device bandwidth.

Unified FPGA processing

Sustain shared functions while reducing redundant data movement.

Host interface remains NVMe-oF. SSDs retain lightweight local mapping and ECC.

SuperSSD Architecture Overview



FPGA data path

TCP / NVMe-oF, shared FTL, aggregation, EC and host I/O.

Embedded ARM

Administration and device management.

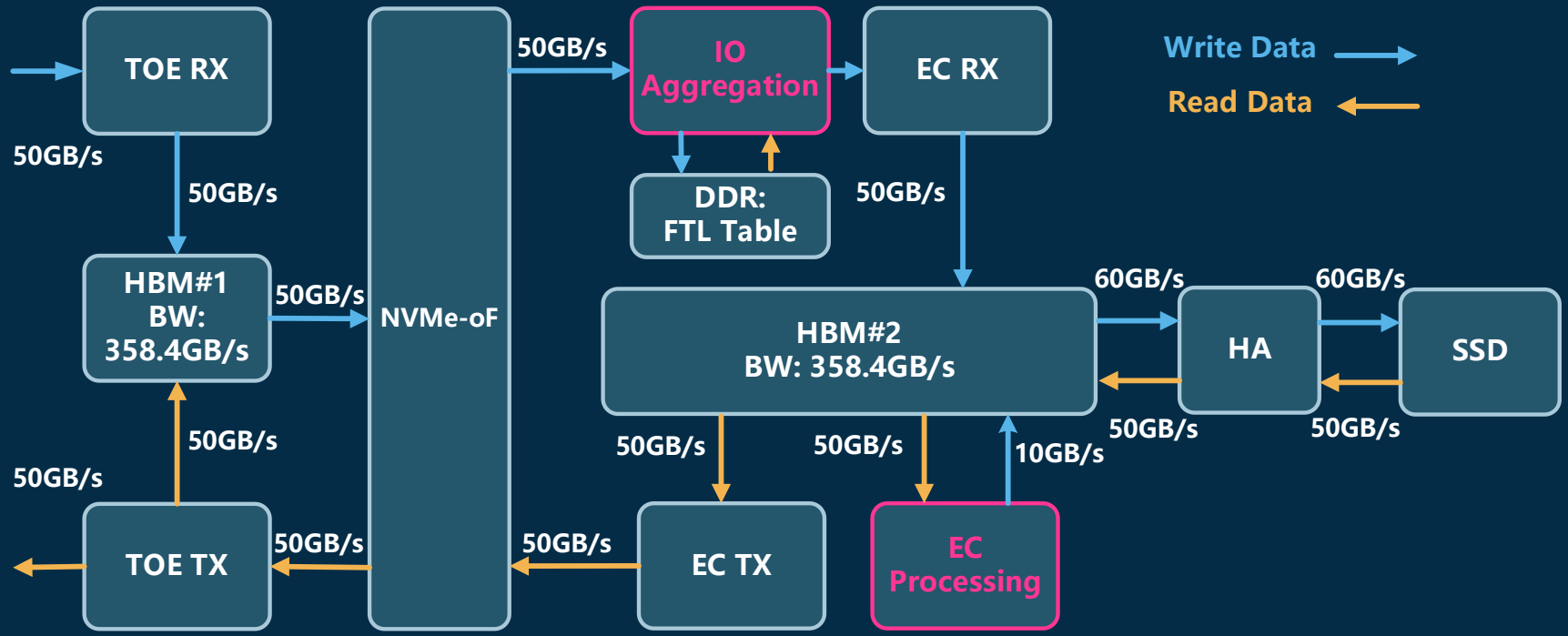
Customized SSDs

Physical addressing and compact local functions.

HBM supports active data processing. DDR holds the shared mapping table.

A Unified Network and Storage Data Path

Challenge: shared processing must keep pace with the network.



Receive and decode

TOE and NVMe-oF process requests through HBM.

Aggregate and encode

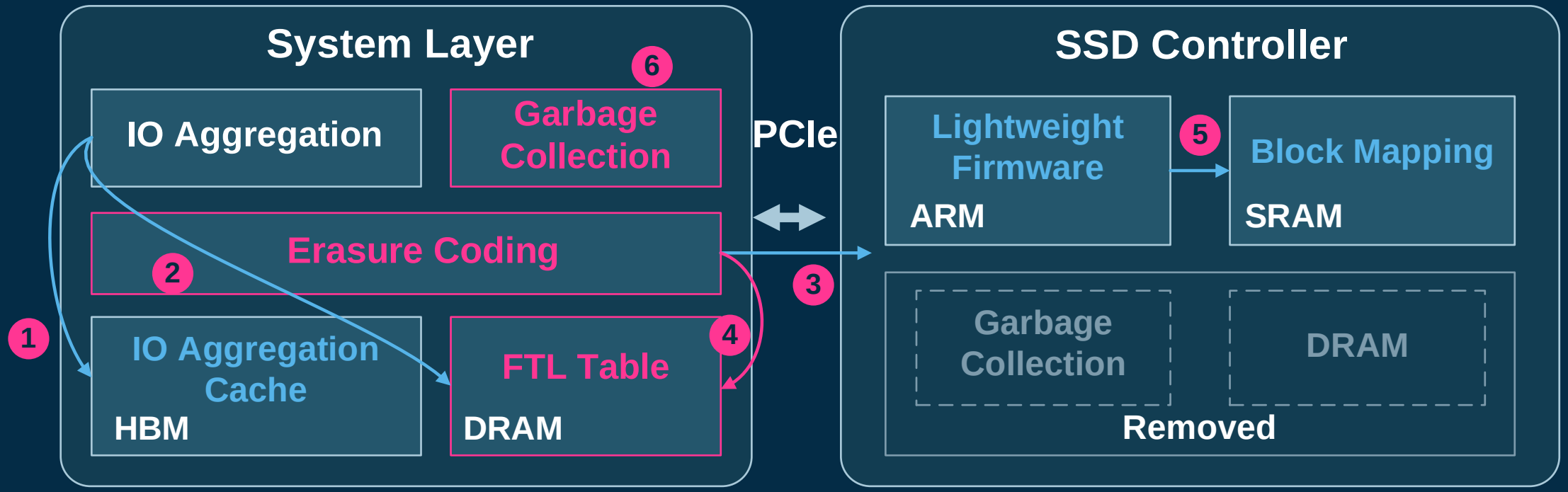
FTL and EC organize the payload within the FPGA path.

Dispatch and return

The host accelerator issues SSD I/O and returns read data.

System-Level FTL and Write Aggregation

Challenge: small random updates make device I/O and parity updates expensive.



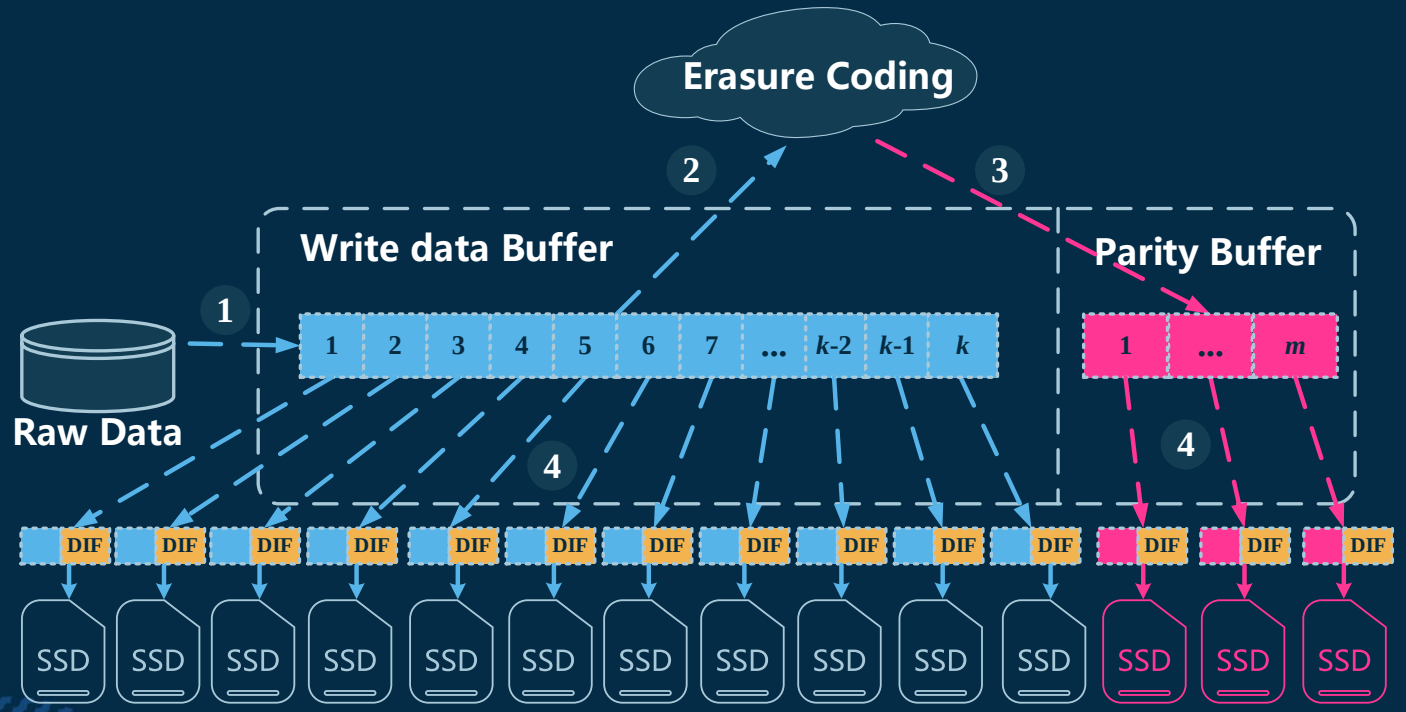
Buffer and remap
Collect unrelated writes and assign new physical locations.

Write full stripes
Full stripes avoid parity read-modify-write.

Reclaim space centrally
Migrate valid data and reclaim obsolete blocks.

System-Level Erasure Coding

Challenge: simpler devices still need strong protection at system level.



k data + *m* parity

Recover up to *m* blocks under coding assumptions.

Integrity checks

Detect corruption with DIF. Reconstruct affected blocks.

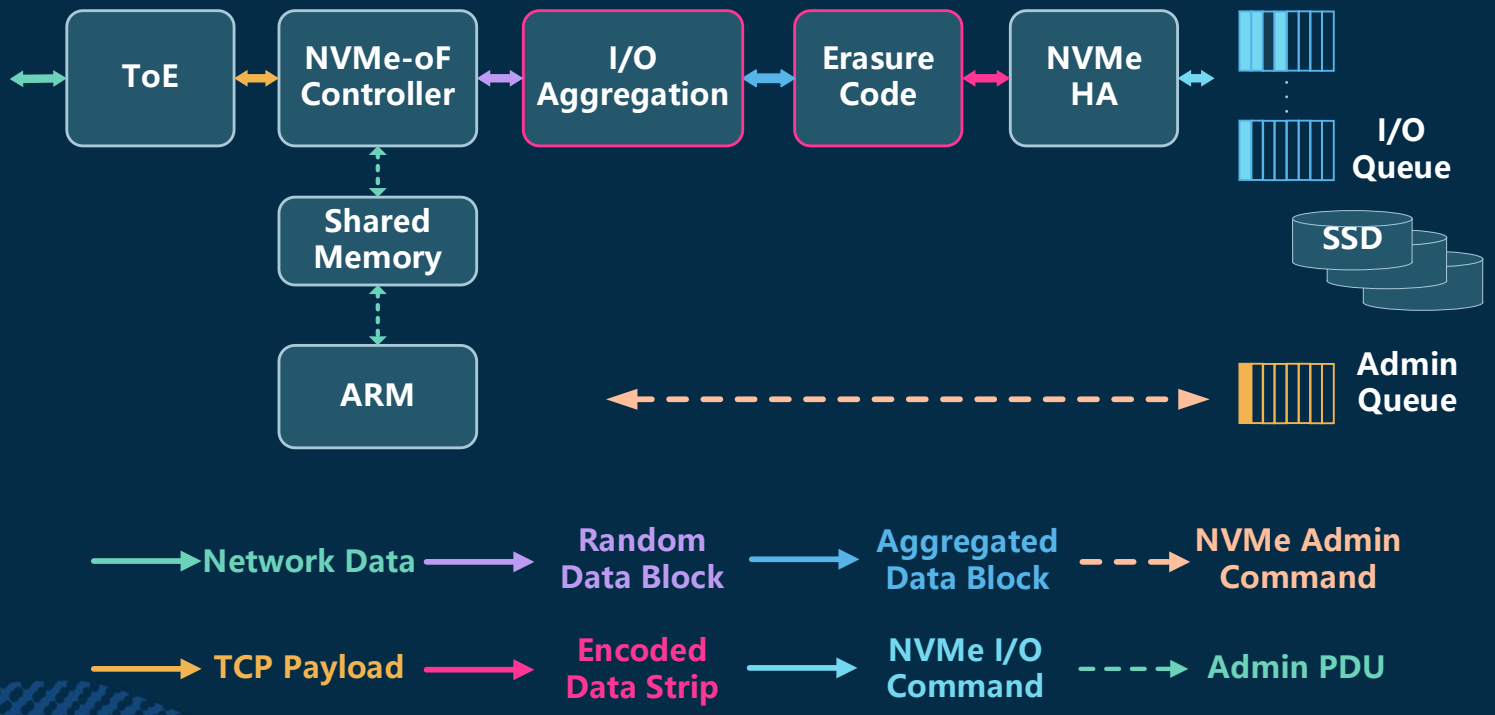
Parallel hardware

Parallel I/O across the group. ~1 µs coding at 300 MHz.

Local SSD ECC remains. Parity consumes 12.5% of raw group capacity.

Data–Control Decoupling

Challenge: preserve control flexibility without loading the fast I/O path.



Data path in FPGA

I/O commands and payloads use SSD I/O queues.

Control path on ARM

ARM software handles fabric and SSD admin commands.

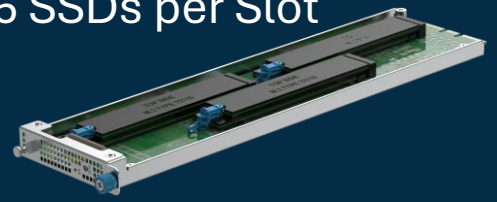
Separate execution

Keep management work away from latency-sensitive transfers.

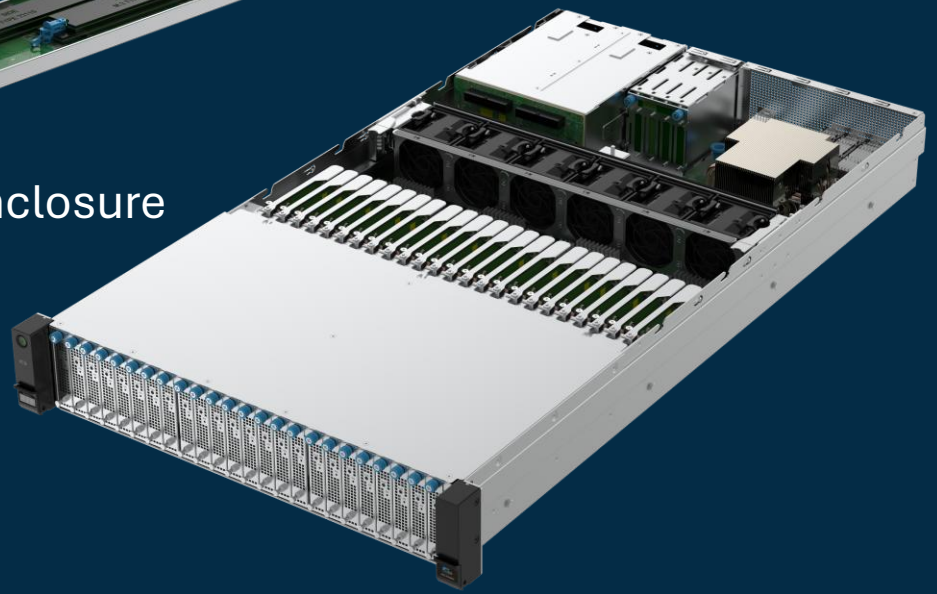
Software retains flexibility for supported management functions.

Hardware Prototype

M.2 SSD Carrier
6 SSDs per Slot



JBOF Enclosure



Implementation

Altera SoC FPGA at 300 MHz
4 ARM Cortex-A53 cores

Measured configuration

64 customized C4000 SSDs
Four 14+2 groups, 200 GbE

Design capability

Up to 144 SSDs via 4 switches
Up to 400 Gbps

Experimental Setup

	CPU-JBOF	DPU emulation	SuperSSD
CPU	32-core Xeon	16-core Xeon	4-core ARM A53
NIC	ConnectX-6	ConnectX-6	FPGA
Memory	128 GB	32 GB	128 GB
Backend	12 × X200P[42]	12 × X200P[42]	64 × C4000[41]
SSD class	Enterprise	Enterprise	Consumer DRAM-less
R/W IOPS	3300K/900K	3300K/900K	80K/29.2K
R/W BW	14000/8200MB/s	14000/8200MB/s	4000/1500MB/s
Erasure coding	N/A	N/A	Four 14+2 groups

Single-drive random-write IOPS: X200P / C4000 = 900K / 29.2K ≈ 30.8×

[41] Hikvision C4000, https://www.techpowerup.com/ssd-specs/hikvision-hiksemi-c4000-512-gb.d1246.
[42] Phison X200P, https://www.phisonenterprise.com/.

Microbenchmarks Explain the Mechanisms

Low-load latency

NVMe-TCP, one thread, QD = 1

Latency	X200P	SuperSSD
Read	93.7 µs	**86 µs**
Write	36.2 µs	**29 µs**

8.2% ↓ read 19.9% ↓ write

Lower latency with the integrated data path.
One X200P versus one SuperSSD EC group.

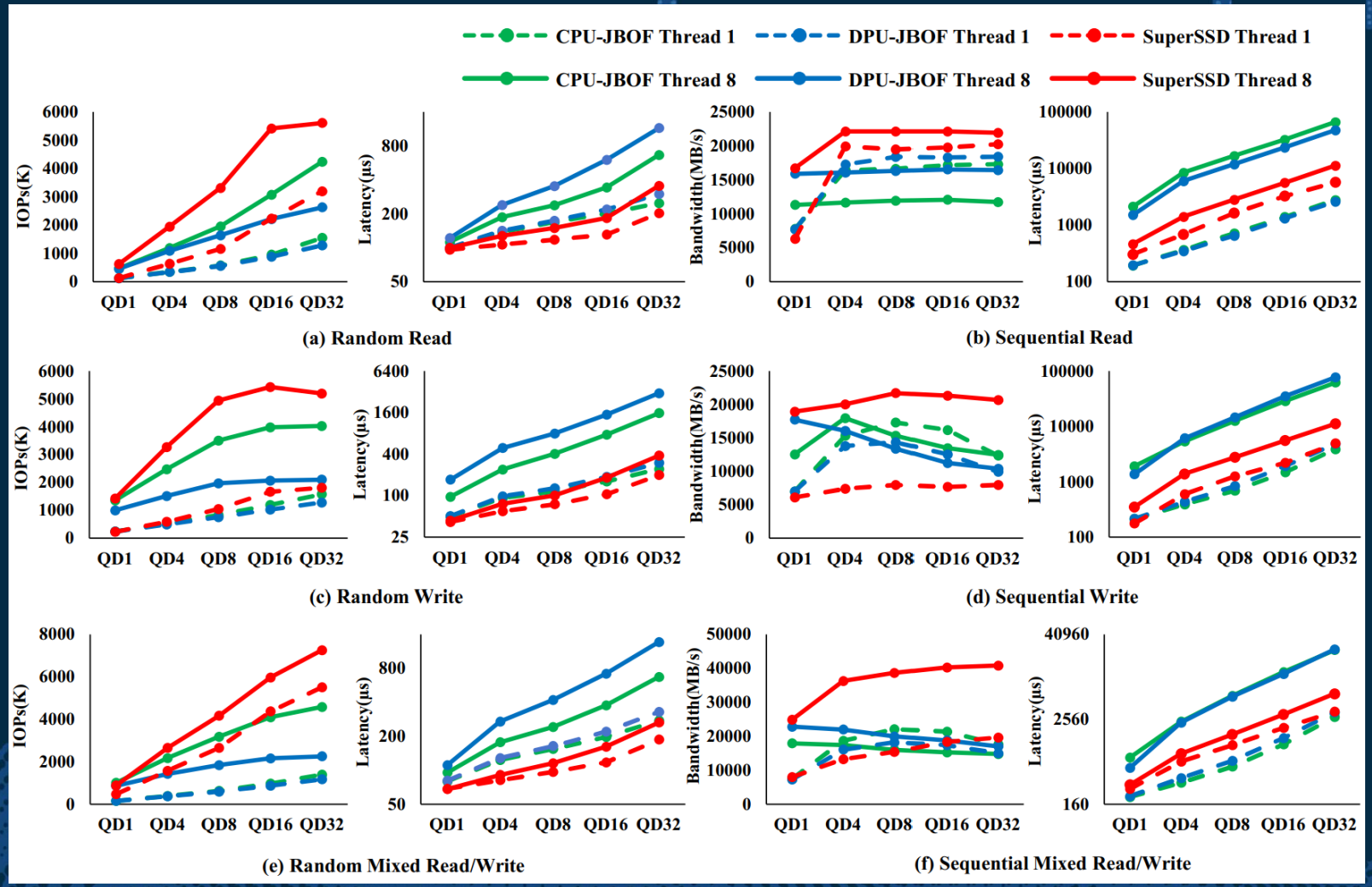
Random-write efficiency

Configuration	Max. IOPS
11+0, no aggregation	213 K
11+1, no aggregation	99.4 K
14+2, FTL-less + aggregation	**2,714 K**

27.3× random-write IOPS

2,714K vs. 99.4K across configurations

Higher Throughput with Lower Latency



1.35×

peak random-write IOPS versus CPU-JBOF

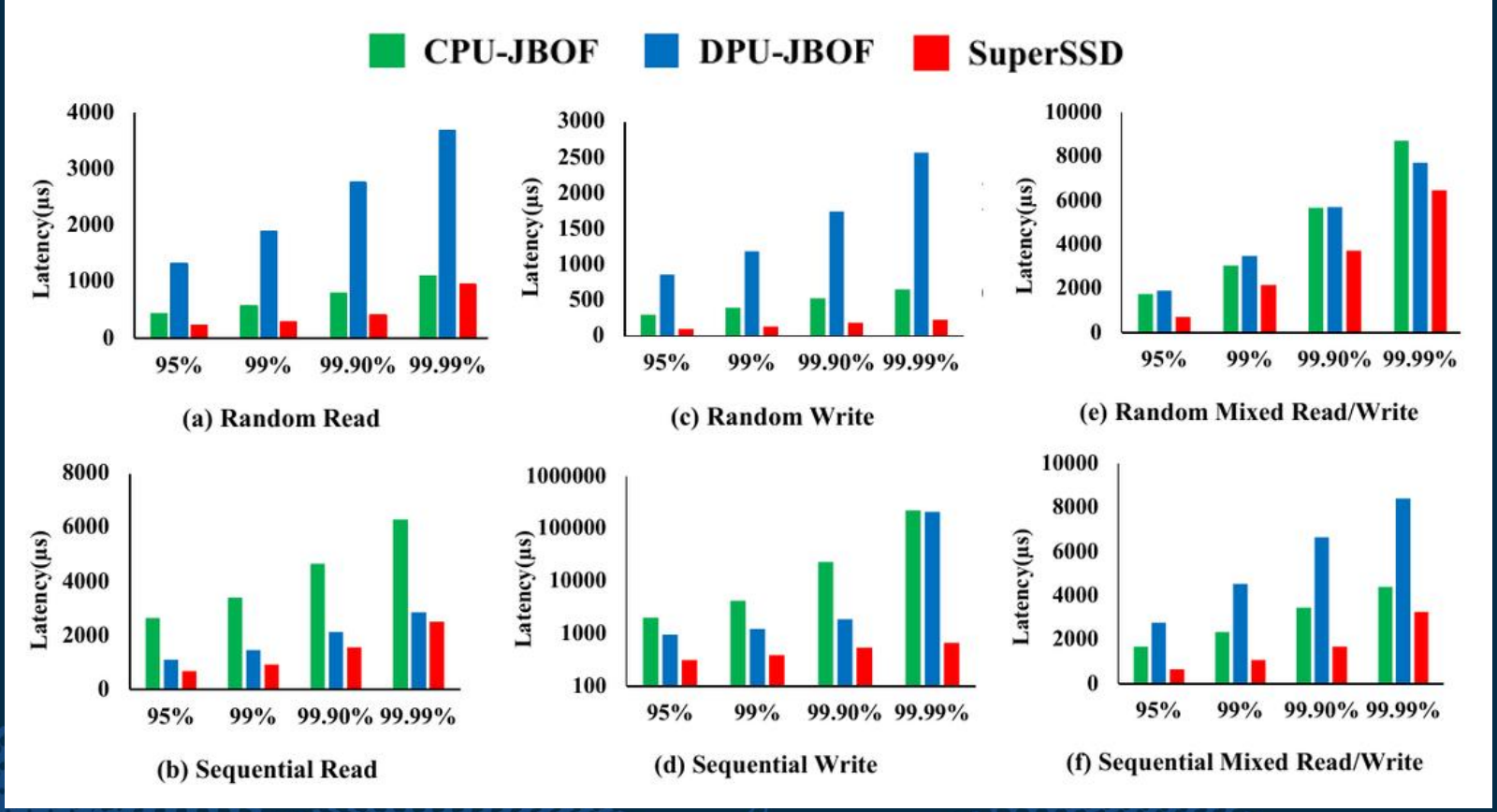
2.50×

versus DPU emulation

SC26

Lower Tail Latency Across I/O Workloads

Matched throughput; 95th–99.99th percentile latency

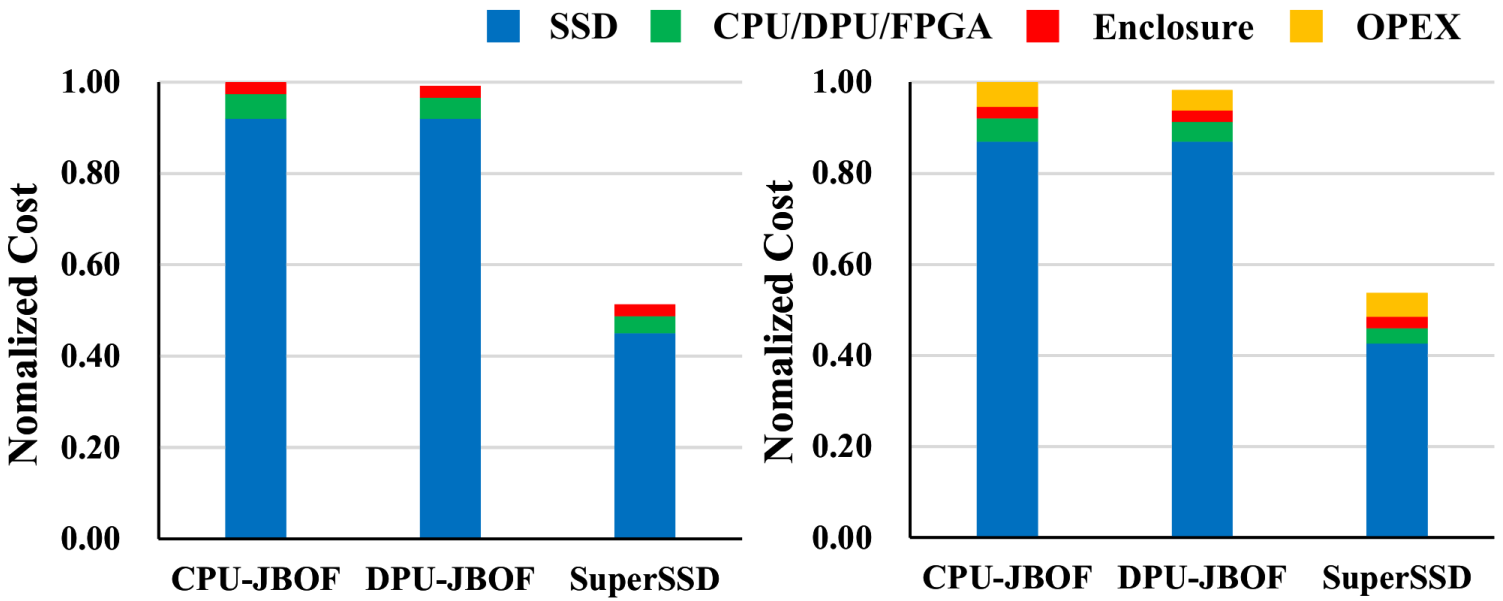


65.95% ↓

lower 99.99th-percentile random-write latency versus CPU-JBOF

Parallel I/O + HBM-buffered write acknowledgment

Lower Modeled Total Cost of Ownership



46.3% ↓

lower five-year TCO vs. CPU-JBOF

5 years, 50% utilization
Power from specifications
Extra replacement and maintenance excluded

SuperSSD: Co-Design the Enclosure as One SSD

The problem

A rigid system–device boundary replicates costly resources in every SSD.

The idea

Treat the entire JBOF enclosure as one coordinated SSD.

Cross-layer co-design centralizes FTL, performance acceleration, and hardware-accelerated erasure-code protection at the enclosure level.

Shared capabilities relax per-drive demands, enabling simpler, lower-cost SSDs.

The result

1.35× peak random-write IOPS; 46.3% lower modeled TCO vs. CPU-JBOF.

Thank You

Q&A

xiaohao@hfut.edu.cn

